Hindawi Publishing Corporation
The Scientific World Journal
Volume 2015, Article ID 123080, 10 pages
http://dx.doi.org/10.1155/2015/123080

Research Article

An Experimental Realization of a Chaos-Based Secure Communication Using Arduino Microcontrollers

Mauricio Zapateiro De la Hoz,[1] Leonardo Acho,[2] and Yolanda Vidal[2]

[1] *Universidade Tecnológica Federal do Paraná, Avenida Alberto Carazzai 1640, 86300-000 Cornélio Procópio, PR, Brazil*
[2] *Control, Dynamics and Applications Group (CoDAlab), Departament de Matemàtica Aplicada III, Universitat Politécnica de Catalunya, d'Urgell 187, E08036 Barcelona, Spain*

Correspondence should be addressed to Mauricio Zapateiro De la Hoz; hoz@utfpr.edu.br

Received 7 May 2015; Revised 27 July 2015; Accepted 9 August 2015

Academic Editor: Chengqing Li

Security and secrecy are some of the important concerns in the communications world. In the last years, several encryption techniques have been proposed in order to improve the secrecy of the information transmitted. Chaos-based encryption techniques are being widely studied as part of the problem because of the highly unpredictable and random-look nature of the chaotic signals. In this paper we propose a digital-based communication system that uses the logistic map which is a mathematically simple model that is chaotic under certain conditions. The input message signal is modulated using a simple Delta modulator and encrypted using a logistic map. The key signal is also encrypted using the same logistic map with different initial conditions. In the receiver side, the binary-coded message is decrypted using the encrypted key signal that is sent through one of the communication channels. The proposed scheme is experimentally tested using Arduino shields which are simple yet powerful development kits that allows for the implementation of the communication system for testing purposes.

1. Introduction

Security and secrecy in communications are some of the most important concerns in societies nowadays. With the advent of worldwide networks and digital communication techniques, the cryptographic techniques that once were restricted to military and state affairs are now covering several domains such as banks, private companies, medical organizations, and so forth. This has led to a very active research field oriented to finding optimal solutions to the problem of communications security [1–3]. As a result, numerous cryptographic techniques that seek to preserve the privacy of the information transmitted have been designed. Chaos is the base of many encryption and decryption techniques because chaotic signals have a highly unpredictable and random-look nature [4].

There are basically two main approaches to designing secure communication systems based on chaotic dynamics: analog and digital. Analog communication systems based on chaos are possible because of the possibility of synchronization [5]. Synchronization occurs when the output of the driving system (master) controls the response system (slave) in such a way that they both oscillate in a synchronized manner. On the other hand, digital chaos communication systems do not depend on chaos synchronization at all. Instead, they usually use one or more chaotic maps in which the initial conditions and the control parameters play the role of the secret key [6].

Several examples of chaos-based communication systems can be found in the literature. For instance, Zapateiro et al. [7] designed a chaotic communication system in which a binary signal is encrypted in the frequency of the sinusoidal term of a chaotic Duffing oscillator. Two chaotic signals of the oscillator are further encrypted with a Delta modulator before they are sent through the channel. In the receiver, a Lyapunov-based observer uses the chaotic signals for retrieving the sinusoidal term that contains the message. A novel frequency estimator is then used to obtain the binary signal.

Furthermore, in a new proposal, Zapateiro De la Hoz et al. [8] investigated a modified Chua chaotic oscillator in which the nonlinear term of the original oscillator was changed for a smooth and bounded function that allows for easier analysis and synchronization with other oscillators. An application to secure communications using the modified oscillator was developed and its performance evaluated by numerical simulations. Hammami [9] proposed an image cryptosystem that makes use of hyperchaotic systems. Synchronization was achieved by assuming some structural assumptions of the master system and using some aggregation techniques associated with the arrow form matrix. Fallahi and Leung [10] developed a chaotic communication system based on a chaos multiplication modulator that encrypts the signal. The chaotic signal is generated by using the Genesio-Tesi chaotic system. The authors also prove that the system security could not be broken with the existing methods at that time. Liu and Sun [11] propose a new design of chaotic cryptosystems in which they use high dimensional chaotic maps along with some cryptography techniques to achieve a high security level. The high dimensionality of the map leads to a high complexity and effective byte confusion and diffusion of the output ciphertext at the time that the small key space problem is overcome. Pareek et al. [12] designed an image encryption scheme in which two logistic maps are used along with an 80-bit key to encrypt/decrypt the images. Eight different types of operation are used to encrypt the pixels of an image; the type of operation is chosen according to the outcome of the logistic maps. This secure communication scheme was criptanalyzed in detail in Li et al. [13]. Lee et al. [14] proposed a chaotic cipher stream, a new scheme for generating pseudorandom numbers based on the composition of chaotic maps. The method consists of using one chaotic map to generate a sequence of pseudorandom bytes and then apply some permutation on them using another chaotic map. Shyamsunder and Kaliyaperumal [15] incorporate the concept of modular arithmetic and chaotic maps for image encryption and decryption. Zhang et al. [16] propose a simple but secure chaotic cipher by improving the familiar permutation-diffusion structure.

Numerous works can be found in the literature that use the logistic map for improving security in communications. The logistic map is a nonlinear discrete map originally used for modeling population growth of different species as well as economic and political phenomena [17–19]. However, under certain conditions it exhibits a chaotic behavior [20]. This characteristic has been exploited in cryptography ever since. For example, Murillo-Escobar et al. [21] presented a symmetric text cipher in which they used a 128-bit secret key, two logistic maps with optimized pseudorandom sequences, plain text characteristics, and one permutation diffusions round. Volos et al. [22] presented a chaotic random bit generator and implemented it in an Arduino board. The microcontroller runs side by side two logistic maps working in different chaotic regimes due to the different initial conditions and system parameters. Statistical tests were carried out to prove security against intruders. Pande and Zambreno [23] presented another experimental realization of a chaotic encryption scheme, this time using a Xilin Virtex 6 FPGA. They implemented a modified logistic map that improves the performance of the logistic map in terms of Lyapunov exponent and uniformity of the bifurcation diagram. Other proposals can be found in Lawrance and Wolff [24]; Chang [25]; and Singh and Sinha [26].

In this paper we present a digital chaos communication system in which the logistic map is used to encrypt the message and key of the transmission. A simple Delta modulator is used along with one of the chaotic maps to encrypt the message. The Delta modulation technique is one of the most simple and robust methods of analog-to-digital (ADC) schemes requiring serial digital communications of analog signals [27]. In this work, the transmitter and receiver are implemented in low cost, small but powerful microcontroller boards: Arduino Uno R3 [28]. The Arduino transmitter receives a message which is analog in nature and encrypts it using a logistic map and the Delta modulator. Then the Arduino receiver decrypts the message and converts it to digital form which corresponds to the Delta-modulated signal. In order to obtain the analog version of the message signal, an analog circuitry performs the demodulation and retrieves the message.

This paper is organized as follows. Section 2 describes the problem to be treated and a scheme of the proposed solution. Section 3 is a brief introduction to the logistic map and its applications to secure communications. Section 4 presents the details of the implementation of the proposed technique. Finally the conclusions are presented in Section 5.

2. Problem Statement

The objective of this paper is to design and implement a communication system to transmit a message $m(t)$ between two points. The goal is to use the logistic map to encrypt the information as a security means. The proposed communication system scheme is shown in Figure 1 and it consists of the following blocks:

(i) *Arduino Transmitter.* This is the core of the transmitter. The Arduino board will take the message $m(t)$ through one of its analog input ports. The Arduino will sample the analog input message, $m(t)$, and convert it to the sampled signal $m(k)$, $k = nT$; T is the sampling time, $n = 0, 1, 2, \ldots$. This signal is then encrypted by using a logistic map and a simple Delta modulator. Afterwards, a key signal, $s(k)$, is generated in order to decrypt the message in the receiver. This key signal is further encrypted using a second logistic map. As a result, the Arduino transmitter generates three outputs: the first one is the encrypted message, $m_e(k)$, the second one is the encrypted key signal, $s_e(k)$, and the third one is an auxiliary key signal, $s_1(k)$, that is used for decryption purposes.

(ii) *Channels.* Three wired channels are used to send the encrypted message and key signals.

(iii) *Arduino Receiver.* This is one of the two main blocks in the receiver side. It takes the signals $m_e(k)$, $s_e(k)$, and $s_1(k)$ to decrypt the Delta-modulated signal before

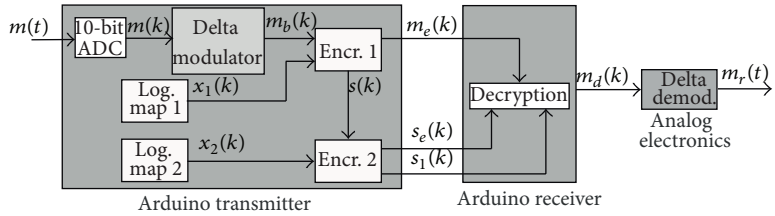

Figure 1: Block diagram of the communication system.

it is converted into its analog form. The output is a digital signal, $m_d(k)$, which corresponds to the signal $m_b(k)$.

(iv) *Delta Demodulator.* This is the second block in the receiver. It is a Delta demodulator consisting of an integrator, a filter, and some amplifiers to retrieve the original message. Its output is a signal $m_r(t)$ that approximates the original signal $m(t)$.

The details of these blocks will be outlined in the following sections of this chapter.

3. The Logistic Map

The logistic map has its origins in the works by the Belgian mathematician Pierre-François Verhulst in the first half of the 18th century [29, 30]. Verhulst published in 1845 and 1847 two articles on how the population growth could be mathematically modeled. He called this model the logistic curve [31, 32], and it is the continuous time version of what nowadays is known as the logistic map.

The logistic map, the discrete-time version of Verhulst's logistic model, is chaotic under certain conditions. Its equation is

$$x_{i+1} = r x_i (1 - x_i), \quad 0 \le x_i \le 1, \tag{1}$$

where r is a constant parameter. Figure 2 is the bifurcation diagram of the logistic map created by varying the parameter r from 2.5 to 4.0.

As can be seen in the bifurcation diagram, there are different regions that depend on the value of r. It is of particular interest when $r = 3$ because there it begins the period doubling that leads to the chaotic dynamics when $r \approx 3.5699\ldots$ until $r = 4.0$. Figure 3 shows the Lyapunov exponent of the logistic map as r is varied from 2.5 to 4.0. It can be seen that the Lyapunov exponent, λ, becomes positive for values of approximately greater than 3.56 which is a strong indicator of chaos [33].

In the next sections, we will use a logistic map as part of an encryption/decryption scheme for transmitting information. We will explain the details of the prototype of this communication system which is implemented on two Arduino Uno boards.

4. Experimental Implementation

4.1. Description of the Communication System. The communication system implemented in this work consists of

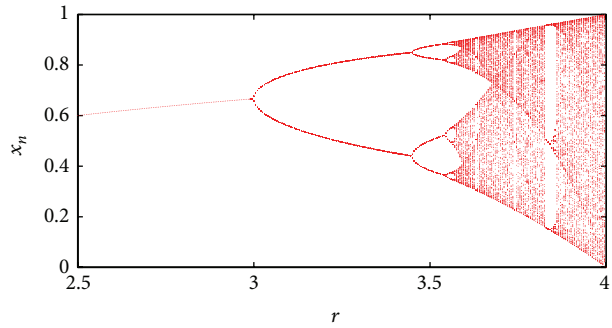

Figure 2: Logistic map bifurcation diagram.

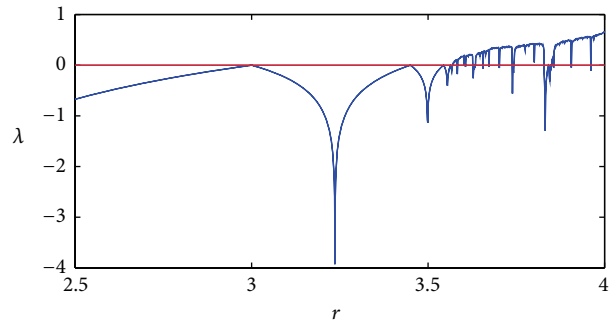

Figure 3: Logistic map Lyapunov exponent.

a transmitter and a receiver whose cores are the Arduino Uno R3 microcontroller boards [28]. These are low cost, simple but powerful microcontrollers based on the ATmega328 chip. They have 14 digital input/output pins (6 of them can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, and a reset button. They can be programmed using a language similar to C++ called Wiring.

The flow diagram of the programs executed by each Arduino is shown in Figures 4 and 5 in order to facilitate the description of the communication system algorithms.

The communication begins when a message $m(t)$, generated by a function generator, and is sent to the analog input A0 of the Arduino transmitter. Arduino analog inputs only accept unipolar signals in the range from 0 V to 5 V. An embedded 10-bit ADC converts the input signal from analog to digital at a maximum rate of 10,000 samples per second. Since the output of the ADC is a value between 0 and 1023 (the ADC resolution), an internal operation to bring it back to the range from 0 V to 5 V is executed. The result is the sampled message signal $m(k)$.

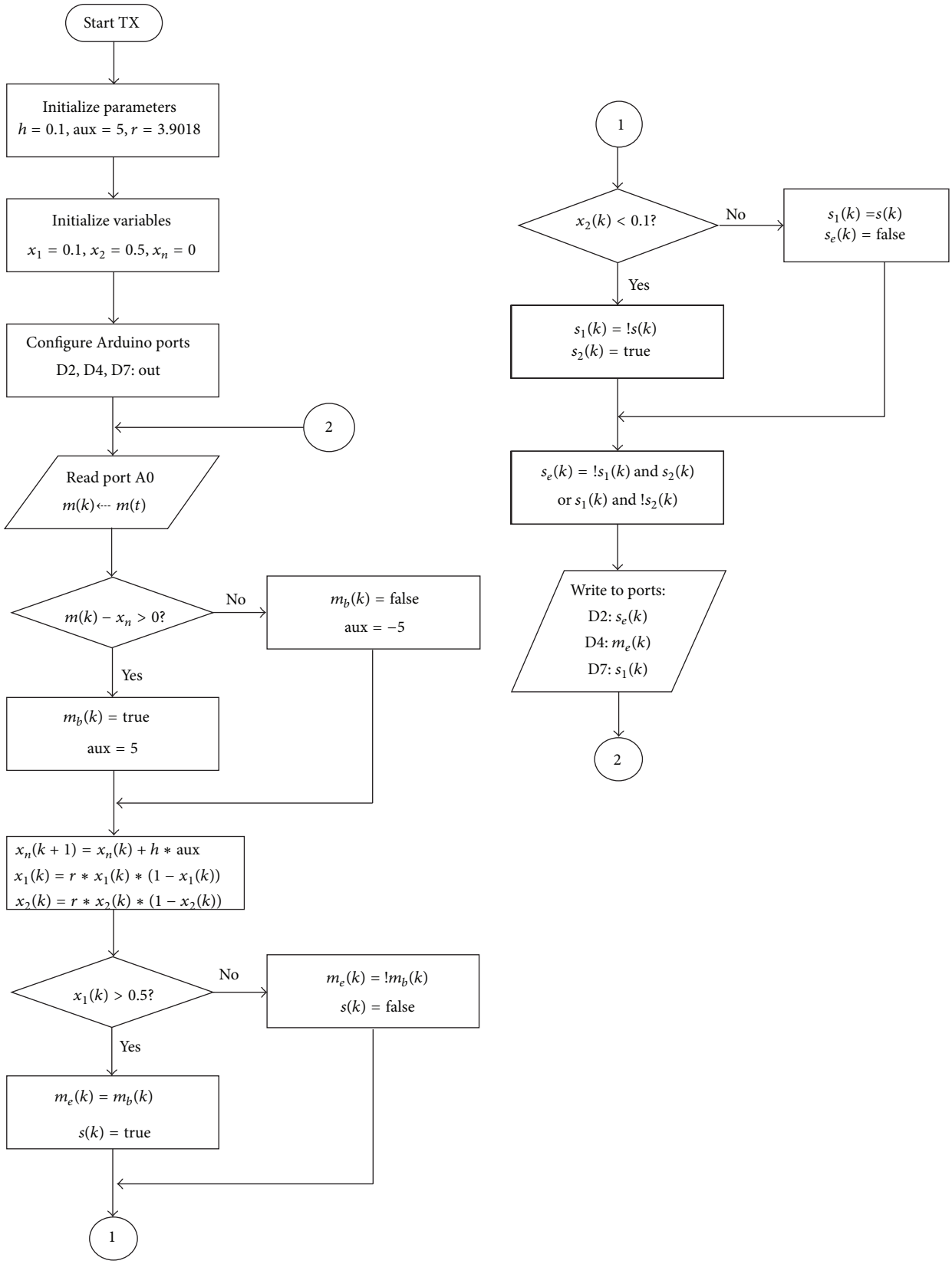

Figure 4: Flow diagram of the transmitter Arduino codes.

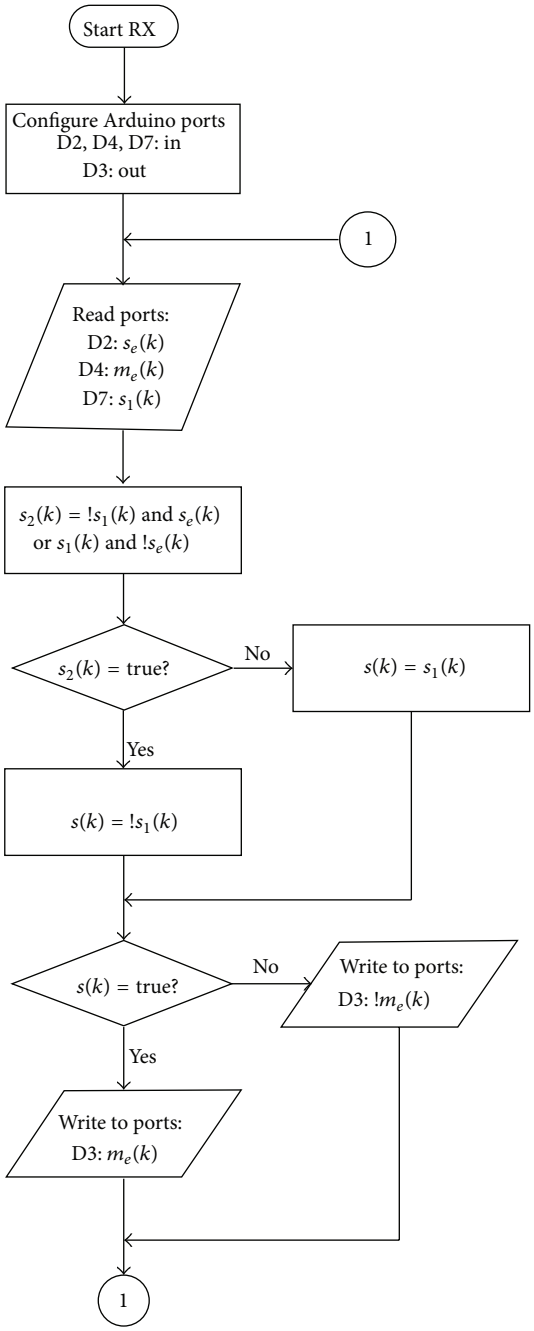

Figure 5: Flow diagram of the receiver Arduino codes.

The next step is the Delta modulation. This kind of modulation can be viewed as an 1-bit ADC conversion scheme since it generates one output bit per input sample. The scheme of the Delta modulation is shown in Figure 6. It consists of a comparator in the forward path and an integrator in the feedback path of a simple control loop. The inputs of the comparator are the signal to be modulated, $m(k)$, and the output of the integrator, $x_n(k)$. As a result, the modulated output, $m_b(k)$, is either true (high) or false (low) at any given time as shown in Figure 7. In this figure we see an input signal and the integral of the expression $m(k) - x_b(k)$.

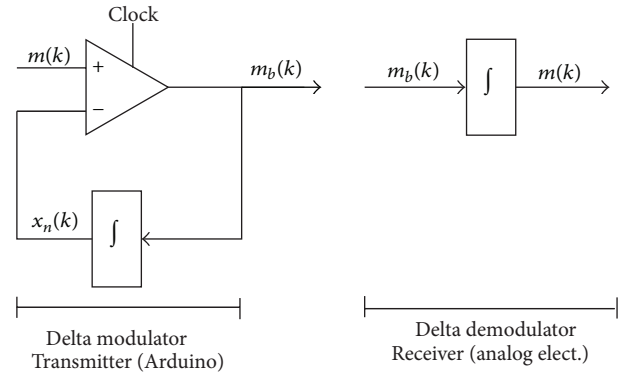

Figure 6: Diagram of the simple Delta modulator.

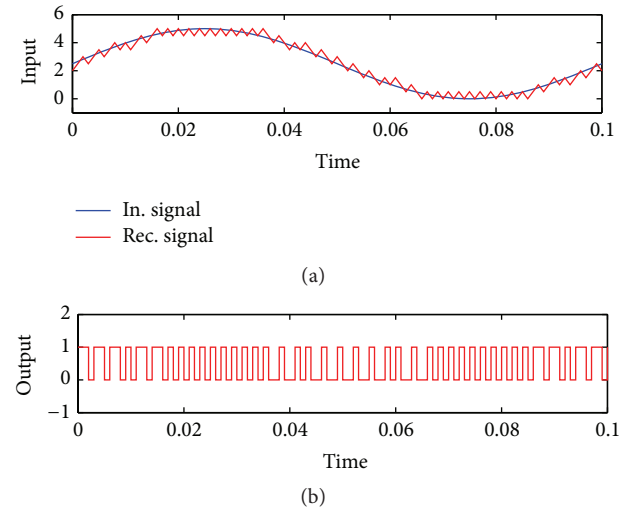

Figure 7: Delta modulation example of a sine input signal. (a) Input signal and reconstructed signal comparison. (b) Modulated output.

For instance, if $m_b(k)$ is ramping up and its output is less than the input, the integrator output will continue ramping up; otherwise it will ramp down. The signal $m_b(k)$ is the differential of the input and thus it can be reconstructed in the receiver by integrating it. In this work, the integral signal, $x_n(k)$, is digitally generated by the Arduino program. On the other hand, the reconstructing integrator of the receiver is implemented with analog electronics as will be explained later. A full description of the Delta modulation technique can be found in Taylor [27].

After one bit from the Delta modulator is obtained, the next step is the message encryption. In order to do so, two logistic maps are called to generate two values $x_1(k)$ and $x_2(k)$. The logistic maps have different initial conditions; that is, $x_1(0) \neq x_2(0)$. Firstly, the message is coded with a value `true` or `false` that is assigned depending on the value $x_1(k)$ of the first chaotic map as can be seen in Algorithm 1 (Part 1), where $m_e(k)$ is the encrypted message and $s(k)$ is the key necessary to retrieve $m_e(k)$.

In order to increase the security of the system, the key, $s(k)$, is further encrypted following the same scheme. It is

```
Part 1
(1) if x1(k) > 0.5 then
(2)      me(k) = mb(k)
(3)      s(k) = true
(4) else
(5)      me(k) = !mb(k)          ▷Symbol ! means boolean negation
(6)      s(k) = false
(7) end if
Part 2
(8) if x2(k) < 0.1 then
(9)      s1(k) = !s(k)           ▷Symbol ! means boolean negation
(10)     s2(k) = true
(11) else
(12)     s1(k) = s(k)
(13)     s2(k) = false
(14) end if
Part 3
(15) se(k) = (!s1(k) AND s2(k)) OR (s1(k) AND !s2(k))
```

Algorithm 1

done by assigning it a value `true` or `false` that depends on the second chaotic map value, $x_2(k)$, as shown in Algorithm 1 (Part 2), where $s_1(k)$ and $s_2(k)$ are auxiliary signals that are used for encrypting and decrypting the key signal $s(k)$.

The key is then finally encrypted by applying the XOR function to the variables $s_1(k)$ and $s_2(k)$ to yield Algorithm 1 (part 3).

The signals $s_e(k)$, $m_e(k)$, and $s_1(k)$ are sent to the receiver through digital outputs D2, D4, and D7, respectively.

In the receiver, the signals $s_e(k)$, $m_e(k)$, and $s_1(k)$ go directly to the Arduino inputs D2, D4, and D7, respectively. The flow diagram of the receiver program is shown in Figure 5 as well. The first step in decrypting the message is the decryption of the key signal $s_e(k)$. This is done by applying the boolean formula that reverts the encryption. The formula to calculate $s_2(k)$ given $s_e(k)$ and $s_1(k)$ is obtained as follows. Recall that in the transmitter $s_e(k)$ is obtained by using the XOR function

$$s_e(k) = s_1'(k) \cdot s_2(k) + s_1(k) \cdot s_2'(k), \tag{2}$$

where $(\cdot)'$ is the complement operation of the corresponding logic variable. The truth table of the function in (2) is shown in Table 1. Thus, given $s_1(k)$ and $s_e(k)$ for obtaining the signal $s_2(k)$ would result in the truth Table 2.

The Karnaugh maps technique [34] was used to find the desired simplified expression for $s_2(k)$. It is a pictorial method in which the truth table of the boolean function to be simplified is represented in a bidimensional form. The boolean variables are arranged according to the Gray code. The terms of the simplified expression are found by grouping 1s or 0s in an optimal way and therefore eliminating unnecessary variables. As a result, the following boolean expression for $s_2(k)$ is obtained:

$$s_2(k) = s_1'(k) \cdot s_e(k) + s_1(k) \cdot s_e'(k). \tag{3}$$

Table 1: Truth table for $s_e(k)$.

$s_1(k)$	$s_2(k)$	$s_e(k)$
0	0	0
0	1	1
1	0	1
1	1	0

Table 2: Truth table for $s_2(k)$.

$s_1(k)$	$s_e(k)$	$s_2(k)$
0	0	0
0	1	1
1	0	0
1	1	1

Once $s_2(k)$ is retrieved, the signal $s(k)$ is obtained with Algorithm 2 (part 1).

The signal $m_e(k)$ is finally decrypted by analyzing the value of $s(k)$ (see Algorithm 2 (part 2)), where $m_d(k)$ is the decrypted signal. The output $m_d(k)$ is sent to the output pin D3 and it goes directly to the Delta demodulator realized with analog electronics using operational amplifiers.

As shown in Figure 6, the Delta demodulation consists of an integrator. The signal is passed through different stages as shown in the circuit diagram of Figure 8. The circuit has three main blocks. The first one, composed of the amplifiers U1 and U2, is a unipolar to bipolar converter. Recall that the Arduino inputs must be unipolar so in the case that the original signals are bipolar they must be recovered to its original form at the output of the Arduino. Thus the signal $m(k) \in [0, 5]$ V is converted to a signal $m(t) \in [-2.5, 2.5]$ V. The second block is composed of amplifiers U3 and U4. They are designed to compute the integral of the input signal. It consists of an integrator that performs the Delta demodulation (U4) and

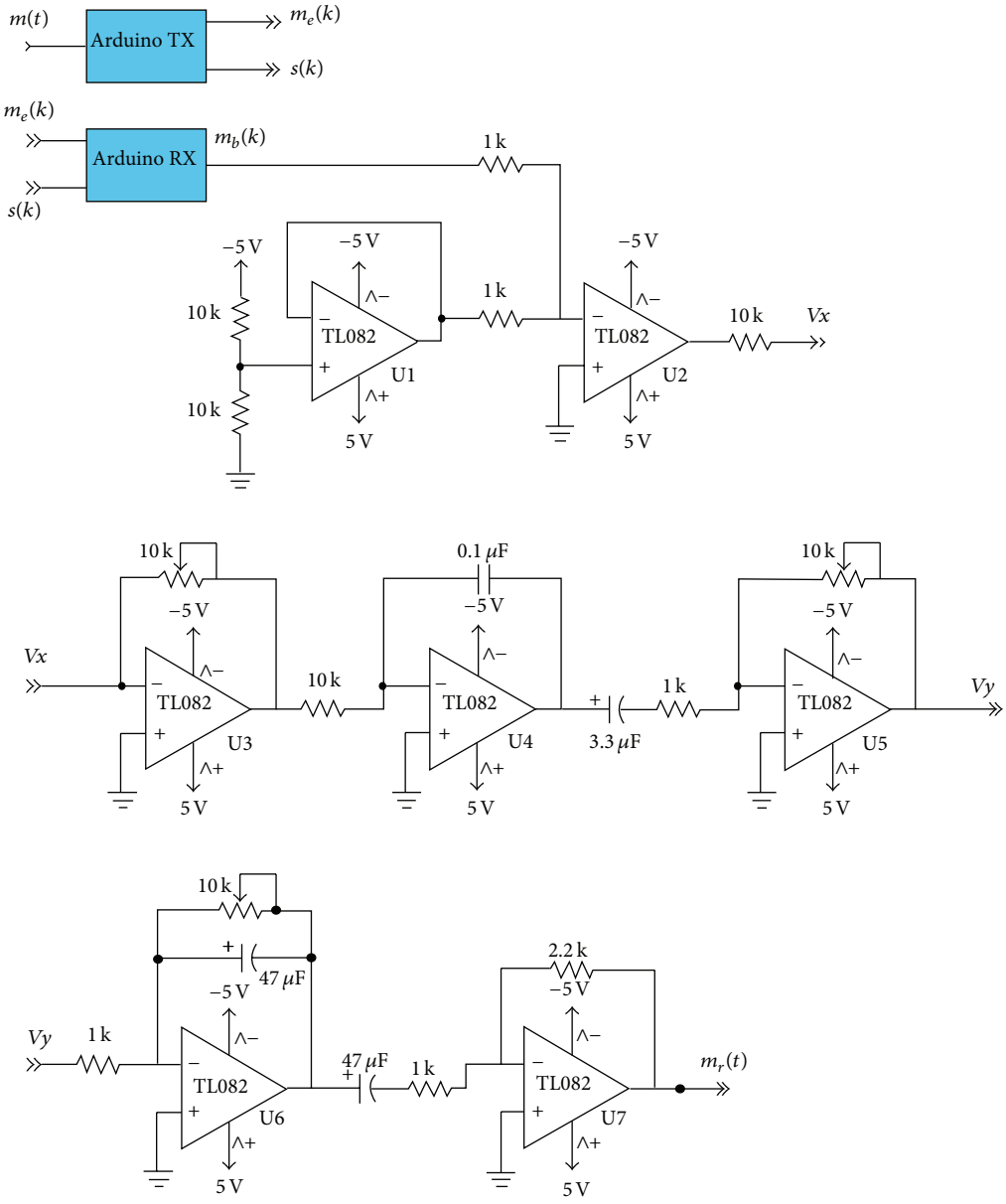

Figure 8: Circuit diagram of the analog electronics in the receiver.

an inverter amplifier (U3) to adjust the quality of its output. These signals are finally sent through a low-pass filter, an amplifier, and an inverter (amplifiers U5–U7) to get the final $m_r(t)$ which should be approximately equal to $m(t)$.

4.2. Experimental Results. For the experiments, the logistic maps were implemented with $r = 3.9018$ and initial conditions $x_1(0) = 0.1$ and $x_2(0) = 0.5$. As an example, the sequence of numbers generated by the logistic map when $x(0) = 0.5$ is shown in Figure 9. Each loop of the transmitter algorithm is executed by the Arduino microcontroller in 210 μs approximately while each receiver loop is executed in 25 μs approximately. This means that the message signal bandwidth should be at most 500 Hz approximately in order to be well retrieved in the receiver.

Figures 10 to 13 are screenshots of the oscilloscope corresponding to the first experiment. In this case, a 125 Hz sine wave, 5 V peak-to-peak amplitude, was used as a message signal. In Figure 10 we see a comparison of the sent message, $m(t)$ (in blue), and the retrieved message, $m_r(t)$ (in yellow). Figure 11 compares the sent message, $m(t)$ (in blue), and the encrypted message signal, $m_e(k)$ (in yellow). Figure 12 is a comparison on the sent message, $m(t)$ (in blue), and the encrypted key signal, $s_e(k)$ (in yellow). Finally, Figure 13 compares the sent message (in blue) and the auxiliary signal $s_1(k)$ (in yellow).

In subsequent experiments, different frequencies and waveforms were tested. Figure 14 shows a 125 Hz triangular wave message (in blue) and its retrieved version (in yellow). Figure 15 compares a 70 Hz sine wave message (in blue) and

Part 1
(1) $s_2(k)$ = (!$s_1(k)$ AND $s_e(k)$) OR ($s_1(k)$ AND !$s_e(k)$)
(2) **if** $s_2(k)$ = true **then**
(3) $s(k) = !s_1(k)$
(4) **else**
(5) $s(k) = s_1(k)$
(6) **end if**
Part 2
(7) **if** $s(k)$ = true **then**
(8) write to port D3: $m_d(k) = m_e(k)$
(9) **else**
(10) write to port D3: $m_d(k) = !m_e(k)$
(11) **end if**

Algorithm 2

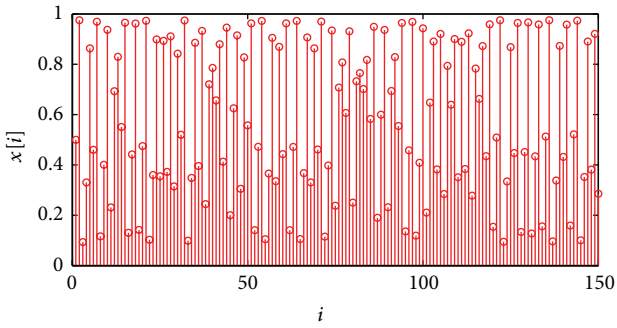

Figure 9: Numbers generated by the logistic map with $r = 3.9018$ and $x(0) = 0.5$.

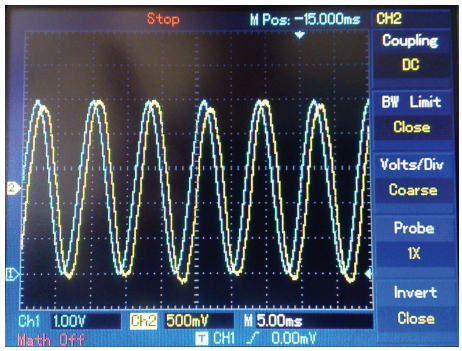

Figure 10: 125 Hz sine wave message. Blue: sent message. Yellow: retrieved message.

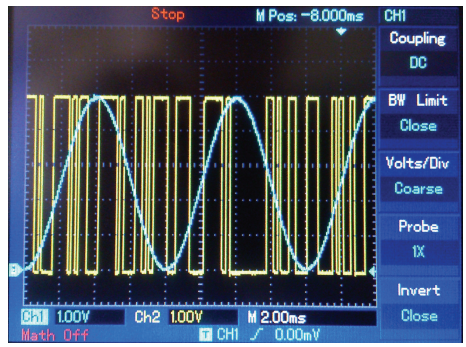

Figure 11: 125 Hz sine wave message. Blue: sent message. Yellow: encrypted message signal.

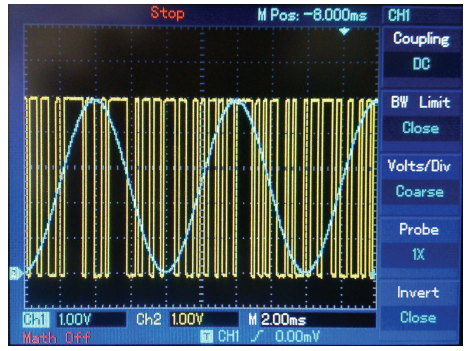

Figure 12: 125 Hz sine wave message. Blue: sent message. Yellow: encrypted key.

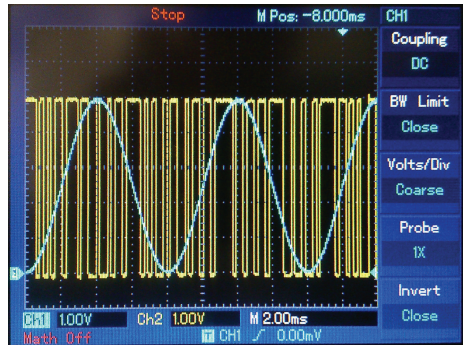

Figure 13: 125 Hz sine wave message. Blue: sent message. Yellow: auxiliary signal $s_1(k)$.

its retrieved version (in yellow). Finally Figure 16 shows a random wave message (in blue) and its retrieved version (in yellow). This signal was generated by making sounds through an *electret* microphone.

5. Conclusion

In this paper we presented a communication system based on chaotic logistic maps and an experimental realization of it. The proposed communication system uses a simple Delta modulator to modulate the message signal and a logistic map for encryption. A key signal is also generated and encrypted in order to retrieve the message in the receiver side without the need for synchronization. The whole system was implemented with Arduino Uno microcontroller boards that run the encryption and decryption algorithms in the transmitter and receiver, respectively. The experiment results showed the feasibility of using the Arduino microprocessors for the task proposed. With the proposed scheme, it is possible to transmit signals whose bandwidth is 500 Hz approximately.

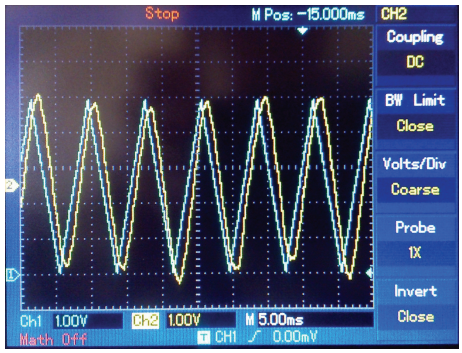

Figure 14: 125 Hz triangular wave message. Blue: sent message. Yellow: retrieved message.

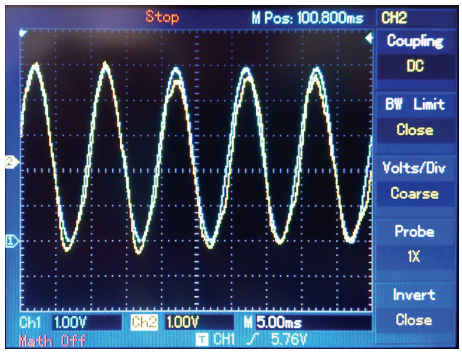

Figure 15: 70 Hz sine wave message. Blue: sent message. Yellow: retrieved message.

Figure 16: Random wave message. Blue: sent message. Yellow: retrieved message.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

First author is supported by the fellowship from CAPES/Programa Nacional de Pós-Doutorado from Brazil. This work was funded by the European Union (European Regional Development Fund) and the Spanish Ministry of Economy and Competitiveness through the research projects DPI2012-32375/FEDER, DPI2011-28033-C03-01, and DPI2014-58427-C2-1-R and by the Government of Catalonia (Spain) through 2014SGR859.

References

[1] L. Larger and J.-P. Goedgebuer, "Encryption using chaotic dynamics for optical telecommunications," *Comptes Rendus Physique*, vol. 5, no. 6, pp. 609–611, 2004.

[2] C. K. Volos, "Chaotic random bit generator realized with a mocrocontroller," *Journal of Computations & Modelling*, vol. 3, no. 4, pp. 115–136, 2013.

[3] C.-K. Chen and C.-L. Lin, "Text encryption using ECG signals with chaotic Logistic map," in *Proceedings of the 5th IEEE Conference on Industrial Electronics and Applications (ICIEA '10)*, pp. 1741–1746, Taichung, Taiwan, June 2010.

[4] L. Kocarev and G. Jakimoski, "Logistic map as a block encryption algorithm," *Physics Letters A*, vol. 289, no. 4-5, pp. 199–206, 2001.

[5] L. M. Pecora and T. L. Carroll, "Synchronization in chaotic systems," *Physical Review Letters*, vol. 64, no. 8, pp. 821–824, 1990.

[6] G. Alvarez and S. Li, "Some basic cryptographic requirements for chaos-based cryptosystems," *International Journal of Bifurcation and Chaos*, vol. 16, no. 8, pp. 2129–2151, 2006.

[7] M. Zapateiro, Y. Vidal, and L. Acho, "A secure communication scheme based on chaotic Duffing oscillators and frequency estimation for the transmission of binary-coded messages," *Communications in Nonlinear Science and Numerical Simulation*, vol. 19, no. 4, pp. 991–1003, 2014.

[8] M. Zapateiro De la Hoz, L. Acho, and Y. Vidal, "A modified Chua chaotic oscillator and its application to secure communications," *Applied Mathematics and Computation*, vol. 247, pp. 712–722, 2014.

[9] S. Hammami, "State feedback-based secure image cryptosystem using hyperchaotic synchronization," *ISA Transactions*, vol. 54, pp. 52–59, 2015.

[10] K. Fallahi and H. Leung, "A chaos secure communication scheme based on multiplication modulation," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 2, pp. 368–383, 2010.

[11] S. T. Liu and F. Y. Sun, "Spatial chaos-based image encryption design," *Science in China, Series G: Physics, Mechanics and Astronomy*, vol. 52, no. 2, pp. 177–183, 2009.

[12] N. K. Pareek, V. Patidar, and K. K. Sud, "Image encryption using chaotic logistic map," *Image and Vision Computing*, vol. 24, no. 9, pp. 926–934, 2006.

[13] C. Li, S. Li, M. Asim, J. Nunez, G. Alvarez, and G. Chen, "On the security defects of an image encryption scheme," *Image and Vision Computing*, vol. 27, no. 9, pp. 1371–1381, 2009.

[14] P.-H. Lee, S.-C. Pei, and Y.-Y. Chen, "Generating chaotic stream ciphers using chaotic systems," *Chinese Journal of Physics*, vol. 41, no. 6, pp. 559–581, 2003.

[15] S. Shyamsunder and G. Kaliyaperumal, "Image encryption and decryption using chaotic maps and modular arithmetic," *The American Journal of Signal Processing*, vol. 1, no. 1, pp. 24–33, 2011.

[16] L. Y. Zhang, X. Hu, Y. Liu, K.-W. Wong, and J. Gan, "A chaotic image encryption scheme owning temp-value feedback," *Communications in Nonlinear Science and Numerical Simulation*, vol. 19, no. 10, pp. 3653–3659, 2014.

[17] C. K. Volos, I. M. Kyprianidis, and I. N. Stouboulos, "The effect of foreign direct investment in economic growth from the perspective of nonlinear dynamics," *Journal of Engineering Science and Technology Review*, vol. 8, no. 1, pp. 1–7, 2015.

[18] J. Miśkiewicz and M. Ausloos, "A logistic map approach to economic cycles. (I). The best adapted companies," *Physica A: Statistical Mechanics and its Applications*, vol. 336, no. 1-2, pp. 206–214, 2004.

[19] D. K. Campbell and G. Mayer-Krees, "Chaos and politics: applications of nonlinear dynamics to socio-political issues," in *The Impact of Chaos on Science and Society*, pp. 18–63, United Nations University Press, 1997.

[20] R. M. May, "Simple mathematical models with very complicated dynamics," *Nature*, vol. 261, no. 5560, pp. 459–467, 1976.

[21] M. A. Murillo-Escobar, F. Abundiz-Pérez, C. Cruz-Hernández, and R. M. López-Gutiérrez, "A novel symmetric text encryption algorithm based on logistic map," in *Proceedings of the International Conference on Communications, Signal Processing and Computers (ICNC '14)*, Honolulu, Hawaii, USA, February 2014.

[22] C. K. Volos, N. Doukas, I. M. Kyprianidis, I. N. Stouboulos, and T. G. Kostis, "Chaotic autonomous mobile robot for military missions," in *Proceedings of the 17th International Conference on Communications*, Rhodes Island, Greece, July 2013.

[23] A. Pande and J. Zambreno, "A chaotic encryption scheme for real-time embedded systems: design and implementation," *Telecommunication Systems*, vol. 52, no. 2, pp. 551–561, 2013.

[24] A. J. Lawrance and R. C. Wolff, "Binary time series generated by chaotic logistic maps," *Stochastics and Dynamics*, vol. 3, no. 4, pp. 529–544, 2003.

[25] S.-M. Chang, "Chaotic generator in digital secure communication," in *Proceedings of the World Congress on Engineering (WCE '09)*, London, UK, July 2009.

[26] N. Singh and A. Sinha, "Chaos-based secure communication system using logistic map," *Optics and Lasers in Engineering*, vol. 48, no. 3, pp. 398–404, 2010.

[27] D. S. Taylor, "Design of continuously variable slope delta modulation communication systems," Motorola Technical Document AN1544, 1996.

[28] Arduino, http://store.arduino.cc/product/A000066.

[29] J. Kint, D. Constales, and A. Vanderbauwhede, "Pierre-François Verhulst's final triumph," in *The Logistic Map and the Route to Chaos*, M. Ausloos and M. Dirickx, Eds., pp. 13–28, Springer, Heidelberg, Germany, 2006.

[30] H. Pastijn, "The logistic map and the route to chaos," in *Chaotic Growth with the Logistic Model of P.-F. Verhulst*, M. Ausloos and M. Dirickx, Eds., p. 3, Springer, Heidelberg, Germany, 2006.

[31] P. F. Verhulst, "Recherches mathématiques sur la loi d'accroissement de la population," *Mémoires de l'Académie Royale des Sciences, des Lettres et des Beaux-Arts de Belgique*, vol. 18, pp. 1–38, 1845.

[32] P. F. Verhulst, "Deuxième mémoire sur la loi d'accroissement de la population," *Mémoires de l'Académie Royale des Sciences, des Lettres et des Beaux-Arts de Belgique*, vol. 20, pp. 1–32, 1847.

[33] A. Wolf, J. B. Swift, H. L. Swinney, and J. A. Vastano, "Determining Lyapunov exponents from a time series," *Physica D. Nonlinear Phenomena*, vol. 16, no. 3, pp. 285–317, 1985.

[34] M. Karnaugh, "The map method for synthesis of combinational logic circuits," *Transactions of the American Institute of Electrical Engineers, Part I: Communications and Electronics*, vol. 72, pp. 593–599, 1953.